

LabVIEW specyfikacja projektu

Automatyczna myjnia

Wersja 1.0

16.04.2021

Skład sekcji 3: Gr. TI3

Mateusz Braszczok

Julia Palichleb

Mateusz Dera

Dominik Oklejewicz

Mateusz Moron

Wstęp

Projekt wykonany w ramach zajęć przedmiotu „Oprogramowanie Systemów Pomiarowych” prowadzonego na Politechnice Śląskiej w Gliwicach w roku akademickim 2020/2021.

Następujący raport przedstawia szczegółową specyfikację wykonanego projektu oraz opis poszczególnych jego funkcjonalności.

Cel projektu

Przedmiotem zajęć było wykonanie zadania programistycznego oraz realizacja projektu myjni samochodowej. Program zrealizowany miał być w środowisku LabView za pomocą szablonu realizującego maszynę stanu. Założeniem projektu jest realizacja systemu obsługującego myjnię samochodową, który wykrywa aktualne położenie samochodu oraz realizuje rutynę automatycznego mycia z uwzględnieniem wyborów dokonanych przez operatora na panelu sterowania. Dodatkowo myjnia ma zapamiętywać poszczególne operacje oraz zapisywać w bazie danych statystyki poszczególnych procesów mycia. Ponad to, dzięki podłączonej kamerze, system może zapamiętywać tablice rejestracyjne obsługiwanych pojazdów oraz aktywować się po naciśnięciu przycisku „start”. Cały system ma zawierać łatwy do użycia panel sterowania, który umożliwi wybranie różnych opcji i procesów mycia dostępnych w myjni. Aplikacja przy pomocy lampek sygnalizuje wykonywanie się poszczególnych procesów oraz podgląd pozostałego do zakończenia wykonywanego procesu czasu. Użytkownik ma również dostęp do suwaka symulującego położenie pojazdów oraz do okien baz danych, gdzie może weryfikować przeszłe zlecenia myjni.

Przegląd systemu

Projekt został stworzony w środowisku LabVIEW 2020 na bazie przykładu z egzaminu CLD pod tytułem “Car Wash” zbudowany na szablonie JKI State Machine, który został dostarczony jako baza projektu. Projekt zbudowany został w całości w warstwie systemowej LabVIEW, bez użycia dodatkowego sprzętu lub narzędzi.

Sprzęt

Do działania program wymagany jest komputer ze specyfikacją wystarczającą do uruchomienia programu LabView 2020.

Do działania systemu wizyjnego wymagana jest kamera z interfejsem szeregowym USB w rozdzielczości Full HD (1920 x 1080 pikseli).

Oprogramowanie

System operacyjny Windows 10 - wersja systemu operacyjnego Microsoft Windows.

LabVIEW (2020) – graficzne środowisko programistyczne stworzone przez National Instruments ułatwiające inżynierom tworzenie aplikacji testowych, pomiarowych oraz do sterowania automatycznego procesami przemysłowymi.

JKI State Machine – szablon maszyny stanów JKI dla programu LabVIEW. Używany jako baza projektu.

SQLite Library - biblioteka procesów, która implementuje niezależny, bez serwerowy silnik bazy danych SQL bez konfiguracji. W naszym programie używamy tej biblioteki do obsługi bazy danych z poziomu aplikacji w LabVIEW.

SQLiteStudio - program do obsługi baz danych. Używany do sprawdzania poprawności wykonywanych poleceń z biblioteki SQLite w programie LabVIEW.

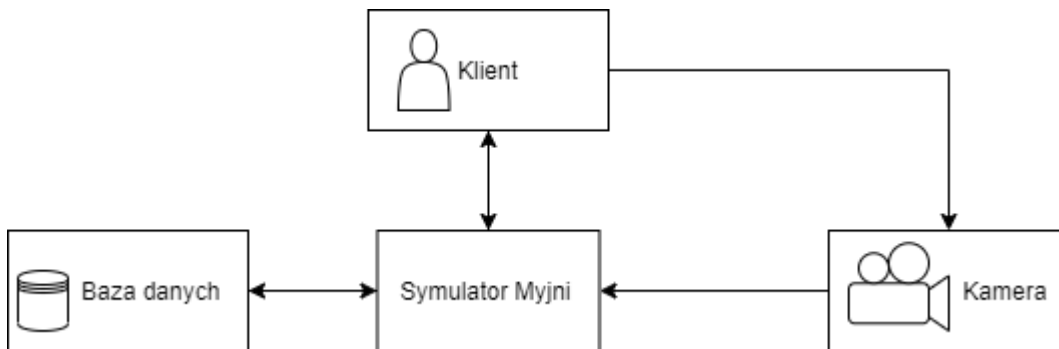
Akwizycja

W programie obraz jest rejestrowany za pomocą kamery Full HD z interfejsem szeregowym USB. Konkretnie jest to część obrazu zdefiniowana jako „Region of Interest” (Obszar zainteresowania). Obraz jest rejestrowany z częstotliwością 10 klatek na sekundę.

Analiza

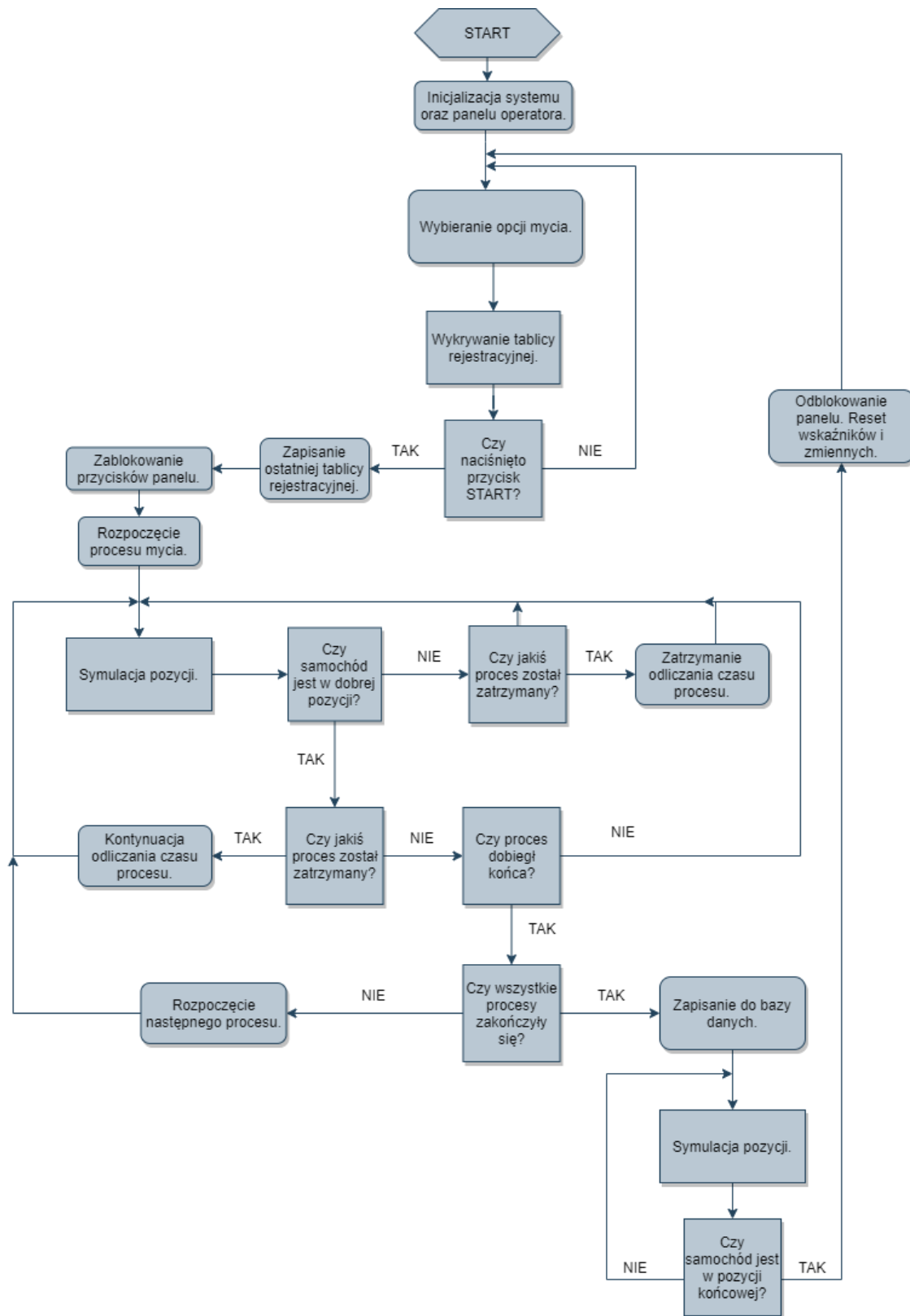
Rozpoznawanie znaków, które pojawią się w obszarze zainteresowania są realizowane przez OCR (optical character recognition) Optyczne rozpoznawanie znaków. By algorytm mógł dobrze rozpoznawać znaki, trzeba najpierw nauczyć go pojedynczych znaków a następnie zapisać w pliku z rozszerzeniem .abc, który potem jest wczytywany w programie.

Schemat przepływu



Rysunek 1 Schemat przepływu

Schemat blokowy

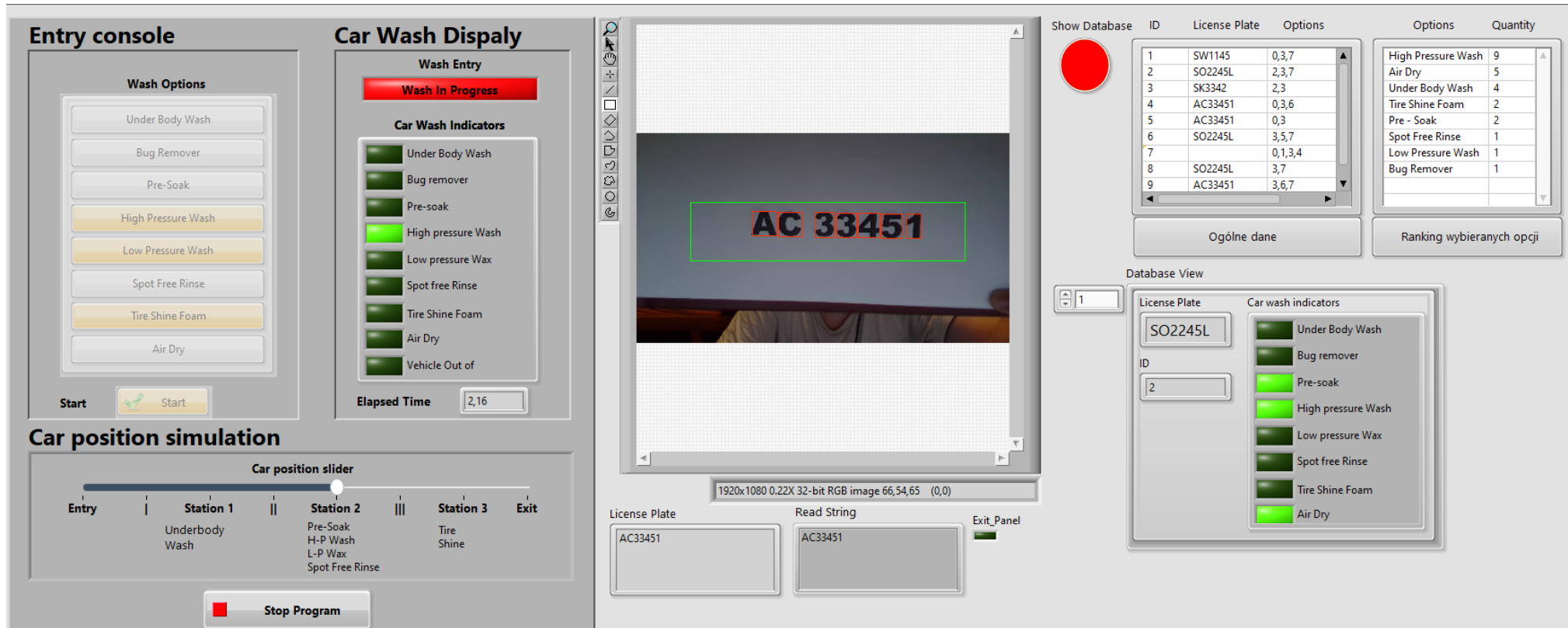


Rysunek 2 Schemat blokowy

Prezentacja

Interfejs użytkownika

Na rysunku pierwszym jest przedstawiony interfejs użytkownika. Po lewej znajduje się panel kontrolny myjni. Zawiera on interaktywne przyciski, lampki sygnalizujące o aktualnym procesie mycia oraz suwak pozwalający na symulację pozycji samochodu. Na środku znajduje się widok z kamery oraz wynik przetworzenia obrazu z systemu wizyjnego. Po prawej stronie użytkownik ma dostęp do interfejsu, który pozwala na analizę przeszłych zleceń myjni.



Rysunek 3 Interfejs użytkownika

Pliki Danych

Podczas działania programu, aktualizowany będzie plik carwash.db, który znajduje się w głównym folderze projektu. Zawiera on dane potrzebne do utworzenia statystyk oraz archiwizacji poszczególnych tablic rejestracyjnych. Plik może zostać otwarty przez program SQLite Studio.

Tabela priorytetów

| Funkcjonalność | Priorytet |
|--|---------------|
| Obsługa przycisków i kontrolek | Krytyczny |
| Symulacja pozycji samochodu | Bardzo Wysoki |
| Komunikaty dla kierowcy | Wysoki |
| Awaryjne wyłączenie programu | Wysoki |
| Odliczanie czasu dla poszczególnych procesów | Wysoki |
| Rozpoznawanie tablic rejestracyjnych za pomocą systemu wizyjnego | Średni |
| Zapisywanie danych do bazy danych | Średni |
| Wizualizacja danych i statystyki | Niski |

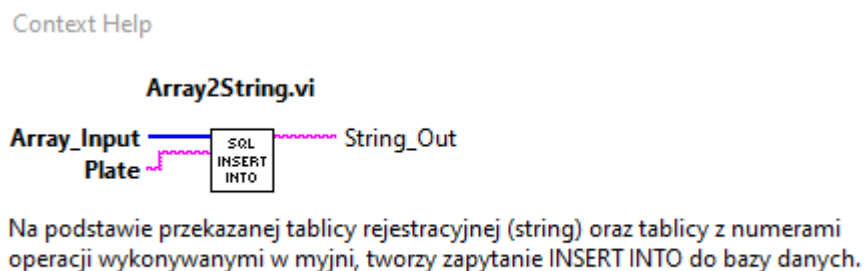
Tabela 1 Tabela priorytetów

Metodologia Testów

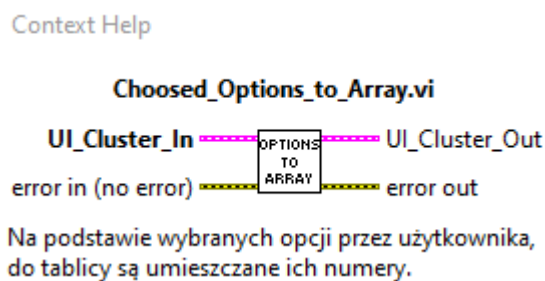
System myjni samochodowej był testowany na różne możliwe warianty zachowawcze od strony użytkownika. System wizyjny został przetestowany na podstawie zbliżenia kartki z tablicą rejestracyjną(Rys. 1). Dodawanie danych do bazy danych zostało przetestowane po przez monitorowanie tworzących się zapytań SQL oraz podgląd na żywo danych znajdujących się w bazie danych. W celu wykrycia wycieków pamięci oraz niezamkniętych referencji użyto programu LabView Desktop Execution Trace Toolkit, błędów i wycieków program nie wykazał.

Opis SubVI's

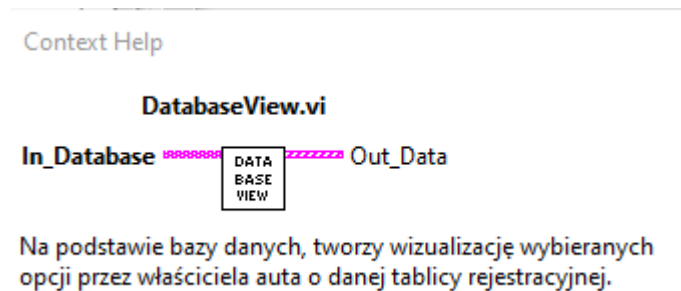
- **Array2String.vi**



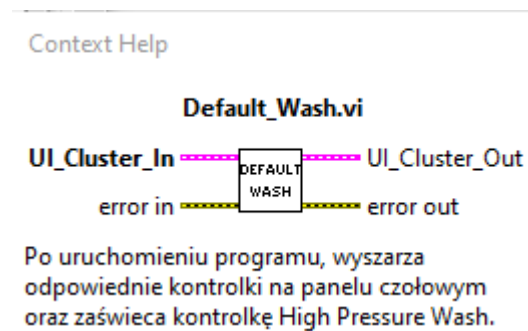
- **Choosed_Options_to_Array.vi**



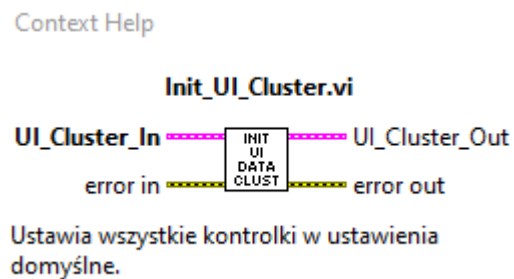
- **DatabaseView.vi**



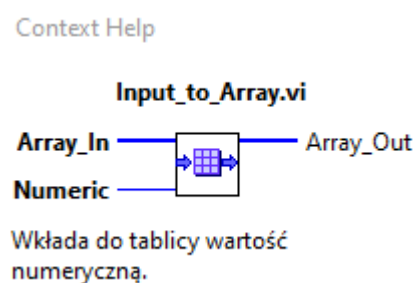
- **DefaultWash.vi**



- **Init_UI_Cluster.vi**



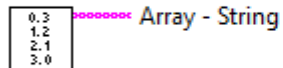
- **Input_to_Array.vi**



- **Options_Statistics.vi**

Context Help

Options_Statistics.vi



Za pomocą polecenia SQL umieszcza w tablicy krotoność wybieranych opcji.

- **Reverse2DArray.vi**

Context Help

Reverse2DArray.vi

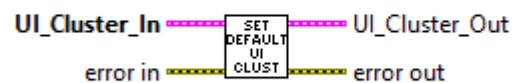


W LabView nie ma wbudowanej opcji do sortowania tablicy w kolejności od największej do najmniejszej.
VI ten służy do odwrócenia tablicy posortowanej w kolejności od najmniejszej do największej.

- **Stop_UI_Cluster.vi**

Context Help

Stop_UI_Cluster.vi

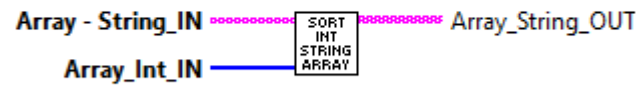


Służy do awaryjnego ustawienia wszystkich kontroltek w stan domyślny.

- **Sort_IntString.vi**

Context Help

Sort_IntString.vi



Sortuje ilość wybranych opcji w formacie INT 32, następnie zamienia te liczby na format String.

Opis Global Variables

- **Bad_position.vi**

Context Help

Bad_position.vi



Przechowuje informację czy pojazd znajduje się na właściwej pozycji.

- **Delay_time.vi**

Context Help

Delay_time.vi



Liczy czas kiedy pojazd jest poza pozycją.

- **Elapsed_time.vi**

Context Help

Elapsed_time.vi



Oblicza całkowity czas gdy nawet pojazd nie jest na poprawnej pozycji.

- **Reset_timer.vi**

Context Help

Reset_timer.vi



Przechowuje informację czy należy zresetować timer.

- **Time_to_elapse.vi**

Context Help

Time_to_elapse.vi



Przechowuje czas, który musi odczekać timer.

Lista i opis stanów

| |
|--------------------------------------|
| "" , "Event Structure" , "Idle" |
| "----- Core -----" |
| Default |
| "Initialize Core Data" |
| "Error Handler" |
| "Exit" |
| "----- Data -----" |
| "Data: Initialize" |
| "Data: Cleanup" |
| "----- UI -----" |
| "UI: Initialize" |
| "UI: Start_Action" |
| "UI: Stop_Action" |
| "UI: Cursor Set" |
| "UI: Front Panel State" |
| "----- Macro -----" |
| "Macro: Wash" |
| "Macro: Initialize" |
| "Macro: Exit" |
| "----- Progress -----" |
| "Progress: Option_choose" |
| "Progress: Set_wait_time" |
| "Progress: Timer" |
| "Progress: Option0" |
| "Progress: Option1" |
| "Progress: Option2" |
| "Progress: Option3" |
| "Progress: Option4" |
| "Progress: Option5" |
| "Progress: Option6" |
| "Progress: Option7" |
| "Progress: Reset variables_to leave" |
| "Progress: Wait for slider" |
| "----- Database -----" |
| "Database: Add to Base" |
| ✓ "Database: Show database" |

- **Event Structure, Idle**

W przypadku wystąpienia zdarzenia wykonuje odpowiednią akcję. Jeżeli kolejka jest pusta to przechodzi w stan bezczynności.

- **Data: Initialize**

Inicjuje wszystkie domyślne wartości zmiennych w klastrze danych.

- **UI: Start_Action**

Po naciśnięciu przycisku start blokuje kontrolki oraz zapisuje wybrane opcje do tablicy.

- **UI: Stop_Action**

W przypadku naciśnięcia przycisku stop przywraca wszystko do ustawień domyślnych.

- **Macro: Wash**

Po naciśnięciu przycisku start, zapisuje stany do kolejki, przez które ma przejść.

- **Progress: Option_choose**

Wstawia do kolejki stany w zależności od wybranych opcji.

- **Progress: Set_wait_time**

Resetuje zmienne globalne powiązane z odmierzaniem czasu oraz zapisuje czas ile powinien stać pojazd na stacji.

- **Progress: Timer**

Po upływie zadanego czasu postoju na odpowiedniej stacji, umożliwia użytkownik przejście do kolejnej części mycia.

- **Progress: Option0 - Option7**

Kontrolują pozycję samochodu dla odpowiednich opcji ustawiając zmienną globalną Bad_position w odpowiedni stan.

- **Progress: Reset variables_to leave**

Resetowanie ikonek na panelu czołowym

Wyzerowanie ikonek na panelu czołowym oraz zmiana komunikatu dla użytkownika.

- **Progress: Wait for slider**

Oczekiwanie na przemieszczenie się użytkownika do wyjazdu.

- **Database: Add to base**

Dodanie kolejnego rekordu do bazy danych z danymi pojazdu, który korzystał z myjni.

- **Database: Show database**

Wyświetla na panelu czołowym informacje z bazy danych o poprzednich użytkownikach myjni.

Wnioski:

W wyniku zajęć laboratoryjnych doprowadziliśmy do końca projekt stworzenia systemu myjni samochodowej, która została wyposażona w system wizyjny oraz bazę danych.

Projekt został pomyślnie doprowadzony do końca oraz z pewnością siebie możemy stwierdzić że zostały spełnione jego najważniejsze założenia. Projekt pozwolił nam na rozwój w kierunku przemysłowych rozwiązań programistycznych. Zgłębiliśmy wiedzę z obsługi środowiska LabView, pakietu JKI State Machine, obsługi baz danych SQLite oraz obsługi prostego systemu wizyjnego.